



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|-------------|-------------------------------|-------------------------|------------------------|
| 10/777,872 | 02/12/2004 | William Preston Alexander III | AUS920030820US1 | 6694 |
| 35525 | 7590 | 10/23/2007 | | |
| IBM CORP (YA) C/O YEE & ASSOCIATES PC P.O. BOX 802333 DALLAS, TX 75380 | | | EXAMINER VU, TUAN A | |
| | | | ART UNIT 2193 | PAPER NUMBER |
| | | | MAIL DATE 10/23/2007 | DELIVERY MODE PAPER |

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/777,872

Applicant(s)

ALEXANDER ET AL.

Examiner

Tuan A. Vu

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 12 February 2004.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 12 February 2004 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date 2/12/04; 7/17/07.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the application filed 2/12/04.

Claims 1-20 have been submitted for examination.

Double Patenting

2. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

3. Claims 5, 15, 20 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 9, 19, 21 of copending Application No. 10,777,743 (hereinafter '743). Although the conflicting claims are not identical, they are not patentably distinct from each other because of the following observations.

Following are but a few examples as to how the certain claims from the instant invention and from the above copending application are conflicting.

As per instant claim 5, '743 claim 9 also recites first call tree for first build and second call tree for second build, subtracting the call tree for the second build from the call tree for the

Art Unit: 2193

first build; outputting the subtracted call tree, wherein for each that exists in both the call tree for the first build and the call tree for the second build, generating a node in the subtracted tree data structure by subtracting a base value of the node in the second tree from a base value of a corresponding node in the first call tree. Although '743 claim 9 does not recite subtracted tree structure identifies differences between the execution of the first build and the second build; the effect of generating a subtracted structure from '743 wherein nodes are call tree nodes, the difference between execution of the respective build for the first and the second tree would have been a obvious limitation.

As per instant claim 15, this is computer medium version of instant claim 5, reciting the same limitations therein, while '743 computer claim 19 corresponds to '743 claim 9; hence '743 claim 19 is a obvious variation of instant claim 15, by virtue of the above analysis.

As per instant claim 20, '743 claim 21 also recites apparatus for generating first call tree for first build and second call tree for second build, subtracting the call tree for the second build from the call tree for the first build; outputting the subtracted call tree. Although the wording of claim 21 is not identical to that of instant claim 20, the subject matter of '743 claim 21 however would be merely an obvious variant of that of instant claim 20.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Specification

4. The disclosure is objected to because some Application identifier format pertaining the 'Related Applications' (Specifications, pg. 1, top) still bears Attorney Docket number, and

Art Unit: 2193

should be updated so they be expressed in proper U.S. Patent Application number format.

Correction is required. See MPEP § 608.01(b).

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 6, 16 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 6 recites 'instructions for creating a node in the subtracting call tree structure having a negative value corresponding to a base value of the node that exists in either of the first call tree ... or second call tree ...'; and one of ordinary skill in the art cannot make use of the above teaching.

According to the specifications, the subtracted node will be set value 2 when there is no corresponding node in the first tree for a corresponding second node tree as the second tree is being incrementally created during the process of call tree traversal (Specs, pg. 6, 2nd para; pg. 35, 1st para). The phrase '*having a negative value corresponding to a base value of the node that exists in either of the first call tree ... or second call tree*' cannot be commensurate with the Specifications, therefore appears to be out of context or obscure in terms of proper and feasible details for implementing the invention. One would not be able to construct a *subtractive tree structure* based on the above incongruous step thus recited. The negative value is treated as a

subtraction from a negative value, as in an addition; or a subtraction where the amount being subtracted from has lesser quantity than the subtracting amount (subtractor).

Correction is required.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made

8. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Levine et al, USPN: 6,349, 406 (hereinafter Levine) in view of Reissman et al, USPubN: 2005/00171818 (hereinafter Reissman).

As per claim 1, Levine discloses a method, in a data processing system, for identifying differences between the execution of a first computer program and a second computer program, comprising:

obtaining a first call tree data structure corresponding to first trace data of an execution of the first computer program (e.g. first pass - Fig. 21; *call stack tree* representation – col. 23, lines 1-8; Fig. 22); obtaining a second call tree data structure corresponding to second trace data of an execution of the second computer program (e.g. second pass – Fig. 21);

subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure (delta event – Fig. 20A; if there are more traces –

Art Unit: 2193

col.18, lines 7-25 – Note: traces to process reads on first trace and second trace with corresponding call tree each); and

outputting the subtracted call tree data structure, wherein the subtracted call tree data structure identifies differences (step 2312 – Fig. 23A; col. 22, line 25-55 – Note: delta time node associated to node of stack tree reads on outputting delta represented by stack tree structure, hence created a *subtracted stack tree* –see step 2314, Fig. 23A) between the execution of the first computer program and the execution of the second computer program.

Levine does not explicitly disclose that the first computer program is a first build thereof and that the second computer program is second build thereof. Levine mentions about using differences that may happens for different platforms, e.g. as Java APIs -- so that trace code has to be used to effect profiling and reduction of code (col. 7, line 61 to col. 8, line 17; col. 9, line 30-48) as endeavored by developers for enhancing code for a given hardware platforms (see BACKGROUND: col. 2, lines 50-58), hence has suggested code improvement made to accommodate software build for platforms. Reissman also discloses a tool to test APIs based on analysis of software builds via a scanning tool whereby dependencies tree is supporting recording of differences between method calls including execution state of these calls, based on such record or dictionary structure so to yield differences (see para 0041, pg. 4; Fig. 3). It would have been obvious for one skill in the art at the time the invention was made to implement the trace analysis and difference structure generating approach by Levine, so that the computer program being traced and converted into tree of events for profile are computer intended for builds as taught by Reissman, because executables intended for different environments (see Reissman) via improved builds can be addressed by profiling and analysis based on events as set

Art Unit: 2193

forth by either Levine and Reissman, whereby based on such differences between trace profiling data, one build as well as testing techniques therefor can be readjusted for improvement over the previous build (see Reismann, BACKGROUND).

As per claim 2, Levine discloses inputting the first trace data and the second trace data to an arcflow tool (Arcflow output – Fig. 17-18), wherein the arcflow tool generates the first call tree data structure and the second call tree data structure based on the first trace data and the second trace data.

As per claim 3, Levine discloses wherein the first call tree data structure and the second call tree data structure are xtree data structures (Fig. 17-18 – Note: tree involving stack data and state reads on eXecution type of tree, i.e. Xtree).

As per claim 4, Levine discloses generating a subtracted call tree data structure includes: copying the first call tree data structure and inserting a pass field (e.g. step 1352, step 1358 -Fig 13) in each node of the first call tree data structure; initializing the pass field (Fig. 13-14; *Base Time* – Fig. 23B) in each node of the first call tree data structure; and walking the second call tree data structure over the first call tree data structure to generate the subtracted call tree data structure (e.g. delta event time - Fig. 20A-B; Fig. 21).

As per claim 5, Levine discloses wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in both the first call tree data structure and the second call tree data structure, generating a node in the subtracted call tree data structure by subtracting a base value (e.g. Fig. 20B; *minus*, step 2312 – Fig. 23A; *Base Time* - Fig. 23B) of the node in the second call tree data structure from a base value of a corresponding node in the first call tree data structure.

As per claim 6, Levine discloses wherein walking the second call tree data structure over the first call tree data structure includes: for each node that exists in only one of the first call tree data structure and the second call tree data structure, creating a node in the subtracted call tree data structure having a negative base value (Note: base value in either first or second call tree reads on delta computing as a result of either adding to a base value; i.e. subtracting a negative value yield a added value – see CUM, Fig. 16 -- OR subtracting from a base value with a greater from the either the first or second tree value - see step 2312 – Fig. 23A; see Fig. 23B, 20B) corresponding to a base value of the node that exists in either of the first call tree data structure or the second call tree data structure.

As per claims 7-8, Levine discloses setting a value of a pass field of the node in the subtracted call tree data structure to a value indicating that both the first call tree data structure and the second call tree data structure contributed to the base value (*delta* – Fig. 20A; 2364 *adjusted base time for routine C*, pg. 23B); setting a value of a pass field of the node in the subtracted call tree data structure to a value indicating that either the first call tree data structure or the second call tree data structure contributed to the base value (sum child self base – Fig. 17; CUM, Fig. 16).

As per claim 9, Levine discloses, wherein nodes in the first call tree data structure and nodes in the second call tree data structure whose paths have not changed between builds are not present in the subtracted call tree data structure (step 2312 – Fig. 23A; col. 22, line 25-55 – Note: delta time being outputted by stack tree structure, hence created a *subtracted stack tree* –see step 2314, Fig. 23A reads on not outputting where there is no delta).

As per claim 10, Levine discloses identifying improvements (e.g. col. 2, lines 37-59; pruning – col. 19, lines 7-30) but does not disclose identifying regressions from the first build to the second build of the computer program based on values associated with nodes in the subtracted call tree data structure. But based on Reissman's same endeavor for tracking dependencies and changes between two builds for improving of code, and implementing test coverage (see Fig. 2, 16-17; para 0040, pg. 4; Fig .13) the regression test for discovery of bugs between two builds is strongly implied. The rationale in claim 1 using Reissman's approach to improve one build over a previous one via test coverage or regression analysis is incorporated herein for the same reasons based on Levine's endeavor and the applicability of code comparison by Reissman, as set forth above.

As per claim 11, Levine discloses a computer program product in a computer readable medium for identifying differences between the execution of a first computer program and a second computer program, comprising:

first instructions for obtaining a first call tree data structure corresponding to first trace data of an execution of the first computer program;

second instructions for obtaining a second call tree data structure corresponding to second trace data of an execution of the second computer program;

third instructions for subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure; and

fourth instructions for outputting the subtracted call tree data structure, wherein the subtracted call tree data structure identifies differences between the execution of the first computer program and the execution of the second computer program;

Art Unit: 2193

all of which step actions having been addressed in claim 1.

Levine does not explicitly disclose that the first computer program is a first build thereof and that second computer program is a second build thereof. But this limitation has been addressed in claim 1.

As per claims 12-19, refer to claims 2-8, 10 respectively.

As per claim 20, Levine discloses an apparatus for identifying differences between the execution of a first computer program and a second computer program, comprising means for:

obtaining a first call tree data structure corresponding to first trace data of an execution of the first computer program;

obtaining a second call tree data structure corresponding to second trace data of an execution of the second computer program;

subtracting the second call tree data structure from the first call tree data structure to generate a subtracted call tree data structure; and

outputting the subtracted call tree data structure, wherein the subtracted call tree data structure identifies differences between the execution of the first computer program and the execution of the second computer program;

all of which step actions having been addressed in claim 1.

Levine does not explicitly disclose that the first computer program is a first build thereof and that second computer program is a second build thereof. But this limitation has been addressed in claim 1.

Conclusion

Art Unit: 2193

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



Tuan A Vu
Patent Examiner,
Art Unit 2193
October 21, 2007